STEREO GRAPHICS!

Paul Stevenson outlines a neat and novel way to get 3-D on a micro

Now that high resolution colour graphics, in various forms, are becoming available in small computers the possibility exists of generating the traditional red/green image pairs necessary to produce a three-dimensional stereoscopic image when viewed through a pair of spectacles with a red filter over one eye and a green over the other.

The geometrical theory behind the positioning in the red and green parts of the image will be given and one example program for the BBC/Acorn computer is included. Finally, possible applications are considered.

Consider first the red and green images which would have to be produced on the page of a book to give the image of a pyramid standing vertically from the page when viewed from 45 degrees. (see Fig 1). This example is chosen because it works particularly well and because the programming is particularly simple when it is later transferred to a VDU screen.

Assume that eye L is to view through a green filter and that eye R looks through a red one. The desired final image is shown in black WXYZA. The eyes are taken as being 6 cm apart. Since a red line on a white ground should not be visible through a red filter and should appear black through a green one and a green line will similarly not be visible through a green filter but will appear black through a red filter, the images of the intended pyramid as seen by each eye separately can be extrapolated back to lines projected onto the horizontal surface as shown. In fact, for simplicity, the only point in the example which needs extrapolating is the apex of the pyramid A, if one assumes that the base of the pyramid lies in the plane of the page. This gives projected points GA and RA as seen by the right and left eyes respectively. Thus all that has to be done to define the green image on the page (screen) is to join GA to the points WXYZ. The red image can then be joined up in a similar manner. In the example the red and green images are symetrically placed with respect to the centre line but this need not be so for other objects.

Fig 2 shows the final appearance of the images on the screen or page. Note that around the base WXYZ, where both red and green images overlie each other, the line should be plotted in black since it has to be present at *both* eyes. The final image received by the brain is in monochrome. Both the BBC and RML 380Z machines make provision for making the logical AND function of the plotted red and green colours equate to black and this is a distinct advantage in the programming.

Fig 3 shows the viewing situation for the example program — it was arranged this way because the final pyramid was required 'correct way up'. (There are other ways of achieving this with the VDU arranged conventionally but the geometry is much harder to explain.) The effect is very powerful if the filters are closely matched to the colours on the screen (more on that later).

Having decided to replace the page with a VDU screen placed horizontally then essentially all that one has to do is to decide on:

a) the size (centimetres) of the pyramid base;
b) the height of the pyramid;
c) the viewing distance and angle (50 cms and 45 degrees in this example);

a) and b) are both variables presented to the program.

What remains in to calculate the screen coordinates of WXY and Z and of the points GA and RA. This involves only simple ratios in similar triangles. At a late stage in the program the position of points GA and RA will have to be converted from centimetres to plottable screen points. This conversion will depend on the resolution of the graphics employed but in the BBC/Acorn machine used this conveniently works out on both axes at about 55 points per centimetre.

The program is listed in Fig 4 and is written in BBC/Acorn Basic but as far as possible it has been kept to a common subset of Basic. For instance, no graphics offset origin or procedure calls been used, both of which would have simplified the code.

Extensions to the program

The following extensions to this simple program are possible if there is a drive to experiment.

First offset the apex of the pyramid relative to the base. Secondly raise the base itself itself off the actual screen. The latter task involves calculating the red and green image positions for all four corners of the base as they will not now overlie each other. The pyramid can now be made to appear suspended above the screen surface (yes it does work) but the height it appears above the screen surface is limited by the screen size you are using — the images start to plot off the top of the screen area if it is raised too high.

A third possibility is to experiment with the viewing angle, but this has not been tried yet. In relation to this, to avoid the silly situation of a VDU monitor lying on its back, a 45 degree viewing angle can be obtained with the VDU in its 'normal' position but with the view standing 45 degrees 'above' the screen. This is more convenient for viewing — the images are plotted 'down the screen' — GA and RA are near the bottom, but the pyramid 'sticks out' towards the keyboard (not so useful for teaching). Talking about teaching use — here is a snag! The effect can only be seen well by one or two people at a time. There appears to be considerable latitude in both viewing distance and angle of view before the effect disappears altogether but at non optimal positions the pyramid is 'distorted'.

Depth illusions

A program has been experimented with, whereby a framework cube is made to appear suspended in the space between the viewer and the screen (see Fig 5).

With appropriate image placing on the screen, the cube can be made to appear, say, halfway between the viewer and the screen. Using the same type of geometry as before the projections of the cube corners onto the screen from each eye can be calculated and then plotted. The effect works but it is (or appears) more difficult to 'conjure up' as it needs the eyes to be focused on empty space at the halfway point where there is nothing there to start with. The main source of light — the screen — acts as a distractor. If the image is on or near the screen everything appears to work much better.

This program has, as input variables,

Lines 40 and 50	fix the pyramid height and base in centimetres.
Line 60	sets a variable to half the base length. The 55 converts from centimetres to screen units.
Line 100	selects the appropriate graphics mode (four colours)
Lines 110 to 116	sets the colours to red and green, with a white background for the graphics.
Line 130	clears the graphics area to its white background.
Line 140	calculates the length of GA to 0 (Fig 1) in cms. The 40.11 is the length of OT (one could almost use 40 cms).
Line 150	calculates the displacement left of the centre line for the point GA in screen units this time.
Line 160	calculates a similar displacement up the screen.
Line 410	choses colour green.
Lines 420 to 510	draws the green image.
Line 655	since the green and red images are symmetrical this sets the displacement right of the centre line for RA. (DU is the same as before.)
Line 660	The clever one! This choses red to plot unless the colour already there is green in which case it draws black. This will make the red/green overlap areas the correct colour.
Lines 670 to 760	plots the red image.
Line 998	waits for any key before restarting.

Program explanation — see fig 4.

the distance of the desired cube from the eyes, its size, and its angle to the horizontal.

Images behind the screen

The principle here is the same as in a) but beware — the relative placings of the red and green images are reversed left for right.

Fig 6 should make this clear and illustrates how a frame cube could be made to appear *inside* the VDU tube. Beware of this red/green reversal effect if you have an object which you wish to intersect the screen. The same difficulties about distraction during focusing apply as in a) — the light from the screen has to be ignored as far as possible.

Selection and rotation of objects

For trial purposes select simple objects. Each corner on an object has to have a projection calculated onto the image plane, and the fewer there are the better. Leave a scale model of the Starship Enterprise till later!

Rotation has been tried with the cube revolving in space between viewer and screen but Basic is too slow to get a nice effect — machine code must be used here.

Practical points

The matching of the filter material bandpass to the intensity/depth of the colour red or green is important. Through even the best red filter some vestige of the red line will be visible on the screen. Reducing overall screen brightness seems to help here. The relative 'blackness' of each colour when viewed through its opposite colour filter should be the same. If one is black and

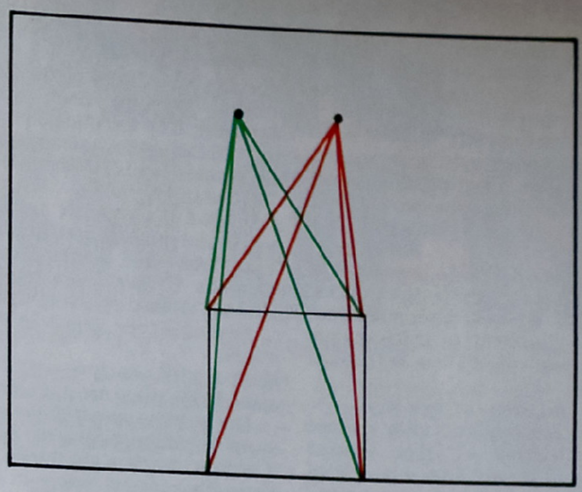

Fig 2. Final images as they appear on the screen or page.

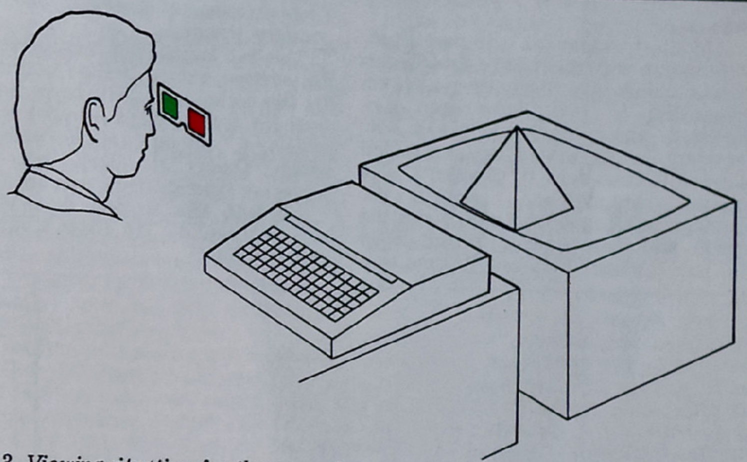

Fig 3. Viewing situation for the example program (put feet on the back of your VDU!!).

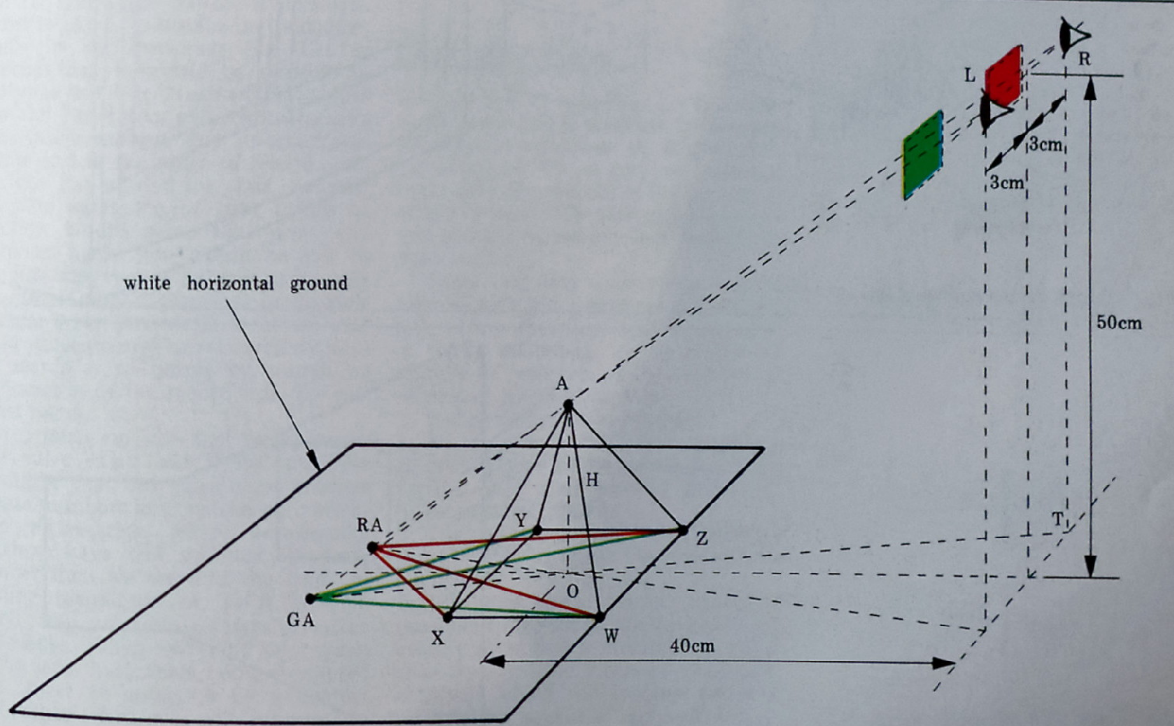

Fig 1. Basic geometry to produce red and green images on a horizontal page or VDU screen.

STEREO GRAPHICS!

the other a shade of grey the brain will not properly 'pull' the stereo image into the middle and one of the red or green images will dominate.

This is where a facility like the RML 380Z's to 'mix' ones own 'shade' of red or green is very valuable. One then accepts whatever filters one can get hold of and fiddles with the colour mix. This is more elegant.

Wire frame models have been used so far as they do not involve the 'hidden line algorithm'. We now have a need for a hidden line algorithm in three dimensions before real objects can be shown. Who will try?

Although rotation of whole objects is slow it is possible to make a shape 'grow' an extension or 'arm' from a fixed base, and on my first example of the vertical pyramid I have had a small 'character' climb one of the sides 'out' of the screen — hurrah for the BBC's (and the 380Z's) programmable characters!

The ideal to aim for is to have a program which will ask for the coordinates of all corners of an object (or read them from disk if they have been prerecorded) and a statement about how they are to be joined up and scaled along differing axes if necessary. The program should then request the required position of the object and the viewer and then procede to plot it on the image plane. This is still some way off as a general package for a wide class of objects.

Conclusion

The impact of seeing one's first stereo 'object' emerging from the screen is immense. I do encourage you to try. To whet the appetite, how about the following developments for the future?

Computerised chess with graphics-generated 3-D pieces (remember the horizontal screen/board in Fig 1).
'Star Trek' and similar games chasing Klingons 'through' the screen into the distance instead of across it.
'Adventure' type games in genuine 3-D mazes.
Maybe with machine code in a dedicated ROM these are not too far off!

On a more prosaic level, and with more educational content, how about:
'Contour' maps in geography.
Three-dimensional graphs in any setting.
Technical drawing projection work — make third angle projection drawings 'come to life'.
Three-dimensional geometry and trigonometry teaching.

This list is probably only scratching the surface but to deflate enthusiasm for the technique a little there is still a need for a convenient or quick way of projecting curved objects onto the screen. Circles are okay, but other curves are more difficult — or are they?

I would like to hear from readers who successfully try these techniques or extend them,.

```
10   REM 3-D PYRAMID P.W. STEVENSON
40   INPUT "PYRAMID HEIGHT (CM)" H
50   INPUT "LENGTH OF BASE (CM)" B
60   D = B*55/2
100  MODE 5
110  VDU 19,2,3,0,0,0
112  VDU 19,131,0,0,0,0
114  VDU 19,128,7,0,0,0
116  VDU 19,130,2,0,0,0
130  CLG
140  X = H * 40.11/(50 - H)
150  DL = 3 * X * 55/40.11
160  DU = 40 * X * 55/40.11
400  REM GREEN IMAGE TO RIGHT EYE
410  GCOL 0,2
420  MOVE 640 - D,0
430  DRAW 640 - D,2 * D
440  DRAW 640 + D,2 * D
450  DRAW 640 + D,0
460  DRAW 640 - D,0
470  DRAW 640 - DL,D + DU
480  DRAW 640 - D,2 * D
490  MOVE 640 + D,0
500  DRAW 640 - DL,D + DU
510  DRAW 640 + D,2 * D
650  REM  RED IMAGE TO LEFT EYE
655  DR = DL
660  GCOL 1,1
670  MOVE 640 - D,0
680  DRAW 640 - D,2 * D
690  DRAW 640 + D,2 * D
700  DRAW 640 + D,0
710  DRAW 640 - D,0
720  DRAW 640 + DR,D + DU
730  DRAW 640 - D,2 * D
740  MOVE 640 + D,0
750  DRAW 640 + DR,D + DU
760  DRAW 640 + D,2 * D
998  INPUT T$
999  GOTO 40
```

Fig 4.

Reference:- *Descriptive Geometry with 3-D Figures* Imre Pal, Hungarian Technical Publishers 1965.

END

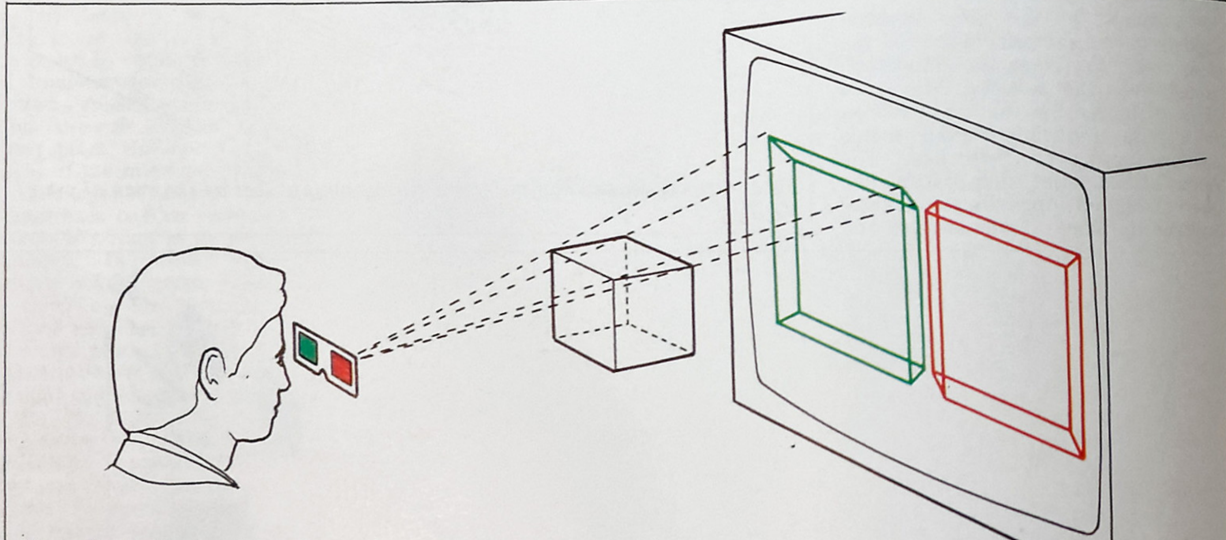

Fig 5. Depth illusion along a line of sight.

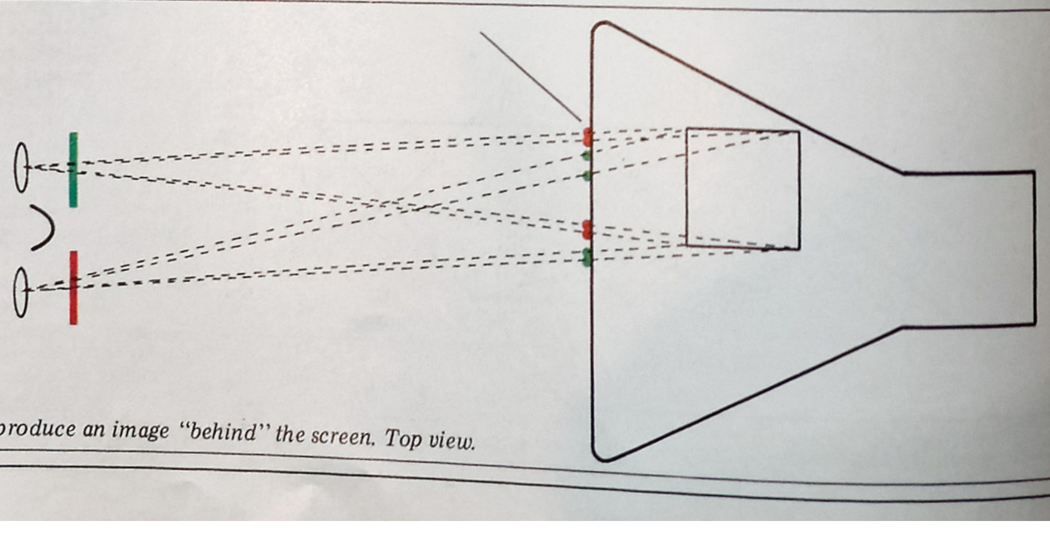

Fig 6. To produce an image "behind" the screen. Top view.